УДК 004

М.Б.Тарасюк
В.О.Денисюк

АЛГОРИТМ ГІПЕРШВИДКОГО СОРТУВАННЯ

Вінницький національний технічний університет

Анотація

Розглянуто алгоритм паралельного гіпершвидкого сортування, який використовує кілька зведених елементів для поділу списку на більше ніж два підсписки. Це дозволяє алгоритму сортувати список ефективніше в певних випадках, особливо коли список уже частково відсортовано або коли він має інші спеціальні властивості.

Ключові слова: алгоритм, паралельний алгоритм, сортування, гіпершвидке сортування, підсписок сортування

Abstract

A parallel hyperfast sort algorithm is considered, which uses several aggregate elements to divide a list into more than two sublists. This allows the algorithm to sort the list more efficiently in certain cases, especially when the list is already partially sorted or when it has other special properties.

Keywords: algorithm, parallel algorithm, sorting, hyperfast sorting, sorting sublist.

Гіпершвидке сортування — це різновид алгоритму швидкого сортування, який використовує кілька зведених елементів для поділу списку на підсписки [1, 2]. Це дозволяє алгоритму сортувати список швидше ніж стандартний алгоритм швидкого сортування, особливо для списків, які вже частково відсортовані або мають деякі інші спеціальні властивості.

У стандартному алгоритмі швидкого сортування один зведений елемент вибирається зі списку, а інші елементи розбиваються на два підсписки залежно від того, чи є вони меншими або більшими за зведений. Потім опорна точка розміщується в остаточному положенні, а два підсписки рекурсивно сортуються за допомогою того самого процесу [1-3].

На відміну від цього, паралельний алгоритм гіпершвидкого сортування використовує кілька зведених елементів для поділу списку на більш ніж два підсписки. Це дозволяє алгоритму сортувати список ефективніше в певних випадках, особливо коли список уже частково відсортовано або коли він має інші спеціальні властивості.

Наприклад, якщо список уже частково відсортовано, використання кількох зведених елементів може дозволити алгоритму пропускати великі розділи списку, які вже знаходяться в правильному порядку, замість того, щоб їх рекурсивно сортувати. Подібним чином, якщо список має деякі інші особливі властивості, такі як велика кількість елементів, які дорівнюють опорній частині, використання кількох опорних елементів може дозволити алгоритму розділити список більш ефективно.

Загалом гіпершвидке сортування є корисним варіантом алгоритму швидкого сортування, який може забезпечити покращену продуктивність певних типів списків. Однак це не завжди найкращий вибір, і стандартний алгоритм швидкого сортування може бути більш ефективним у деяких випадках.

Алгоритм працює таким чином (рис.1):

1. Список розміром **n** розділено на **m** процеси. Припустимо, що список розміром **16** розділено на **4** процеси, кожен процес оброблятиме **4** елементи.
2. Процес серед чотирьох, відповідальних за пошук опорного елемента, знаходить опорний елемент і передає його всім процесам, які сортують свої підсписки послідовно за допомогою широкомовного опорного елемента. Цей крок покращить шанси знайти опорні точки, близькі до справжньої медіани.
3. Повторюємо кроки з попереднього підходу: вибір опорного елемента і трансляція на партнерські процеси, поділ підсписку низьких і високих значень, обмін значеннями між партнерськими процесами.
4. Залишок верхньої половини від одного партнерського процесу та отримана верхня половина від

іншого партнерського процесу об’єднуються в локальний підсписок для кожного процесу.

5. Використовуємо рекурсію для верхньої та нижньої половини кожного підпроцесу, щоб створити відсортований список.
6. Нарешті об’єднуємо процеси, щоб отримати повністю відсортований список.

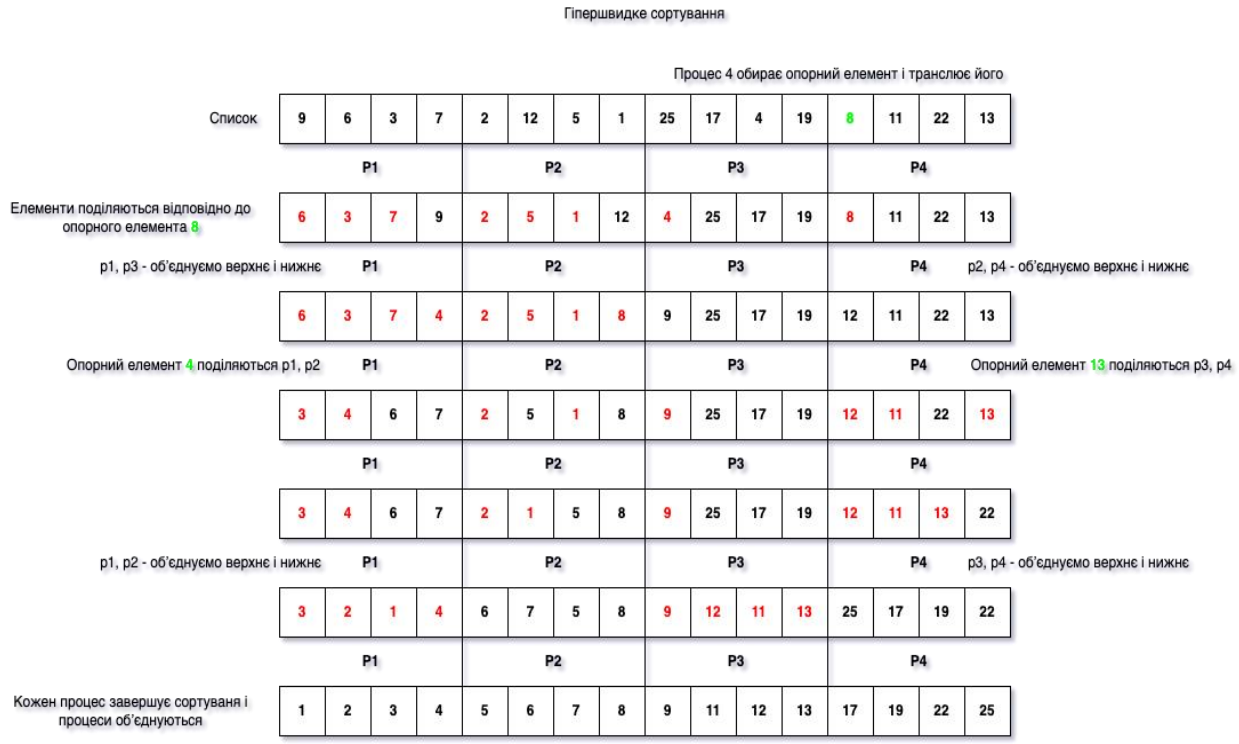

Рисунок 1 – Алгоритм гіпершвидкого сортування.

Особливо треба відмітити той факт, що існують часові витрати на зв’язок між процесами, коли значення передаються між процесами-партнерами, та можливість виникнення дисбалансу навантаження на процеси, але алгоритм кращий порівняно з існуючими, які набагато гірше балансують навантаження. Аналіз алгоритму показав, що існує **log(n)** кроків і **n** процесів, загальна складність розробленого алгоритму за часом становить **θ(nlog(n)), с**кладність простору дорівнює **O(log(n)).**

Висновок

Розглянутий алгоритм паралельного гіпершвидкого сортування використовує кілька зведених елементів для поділу списку на більше ніж два підсписки та дозволяє сортувати список ефективніше в певних випадках, особливо коли список вже частково відсортовано або коли він має інші спеціальні властивості.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Швидке сортування [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/QuickSort.
2. Багатовимірні масиви [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/multidimensional-arrays.
3. Parallel Quicksort [Електронний ресурс]. – Режим доступу: https://iq.opengenus.org/parallel-quicksort

Тарасюк Микола Борисович, студент, група КН-22мс, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, Україна.

Денисюк Валерій Олександрович, к.т.н., доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, Вінниця, Україна, e-mail: vad64@i.ua.

Tarasiuk Mykola Borisovich, student, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Agrarian University, Vinnytsia, Ukraine.

Denysiuk Valerii Olexandrovich, PhD, assistant professor of Computer Sciences Department, Vinnytsia National Agrarian University, Vinnytsia, Ukraine, e-mail: vad64@i.ua.